

《C/C++ 学习指南》

第24.2讲：深度拷贝

作者：邵发 QQ群：417024631

官网：<http://www.afanihao.cn/>

习题：<http://www.afanihao.cn/kbase/>

本书配套纸质教材，具体信息见官网。

C/C++学习指南 邵发 www.afanihao.cn

深度拷贝

当以下情况发生时，需要添加拷贝构造函数。示例：

```
class Text
{
public:
    Text(const char* str)
    {
        // 申请一块内存，保存此字符串
        m_size = strlen(str) + 1;
        m_buf = new char[m_size];
        strcpy(m_buf, str);
    }
    ~Text()
    {
        // 释放此字符串
        delete [] m_buf;
    }
private:
    int m_size;
    char* m_buf;
};
```

深度拷贝

```
int main()
{
    // 定义第一个对象
    Text t1("helloworld");

    // 第二个对象以t1为蓝本进行拷贝
    Text t2(t1);

    return 0;
}
```

深度拷贝

出什么会出错？

// 对象创建

- (1) 对象t1.m_buf, 指向一块内存
- (2) 对象t2拷贝了t1, t2.m_buf指向了同一块内存

// 对象析构

- (1) 对象t1析构, delete [] m_buf;
- (2) 对象t2析构, delete [] m_buf; 出错, 此块内存已经被 delete

错误的根本原因：应该拷贝其数据，而不是拷贝其指针。

解决办法

(1) 正规解决方法

添加拷贝构造函数，拷贝其具体的数据

```
Text(const Text& other)
{
    m_size = other.m_size;
    m_buf = new char[m_size];
    strcpy(m_buf, other.m_buf);
}
```

此种情况称为“深度拷贝”

解决办法

(2) 省事的办法

禁止用户进行拷贝构造，将拷贝构造函数设定为private。

```
private:
    Text(const Text& other)
    {
    }
```

C/C++学习指南 邵发 www.afanihao.cn

小结

只有在需要深度拷贝的情况下，才需要添加拷贝构造函数。